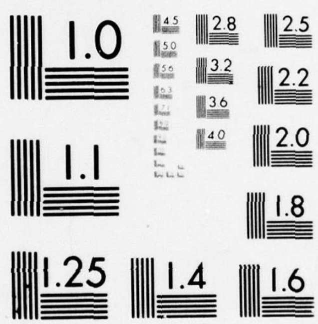
AD-A049 477 STANFORD UNIV CALIF SYSTEMS OPTIMIZATION LAB F/G 12/1
A HYBRID APPROACH TO MULTI-STAGE LINEAR PROGRAMS.(U)
SEP 77 J K HO, J A TOMLIN N00014-75-C-0865
UNCLASSIFIED SOL-77-27 NL

1 OF 1
AD
A049477

END
DATE
FILMED
3-78
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A049477

(11) 2

AD No. ___
DDC FILE COPY

A HYBRID APPROACH TO MULTI-STAGE LINEAR PROGRAMS

BY

JAMES K. HO and JOHN A. TOMLIN

TECHNICAL REPORT SOL 77-27

SEPTEMBER 1977

Systems Optimization Laboratory

Department of Operations Research

Stanford University

Stanford California 94305

see 1473 in back

A HYBRID APPROACH TO MULTI-STAGE LINEAR PROGRAMS

by

James K. Ho[†] and John A. Tomlin[‡]

TECHNICAL REPORT SOL 77-27

September 1977

SYSTEMS OPTIMIZATION LABORATORY

DEPARTMENT OF OPERATIONS RESEARCH

Stanford University
Stanford, California

[†] Brookhaven National Laboratory, Department of Applied Mathematics, Upton, L.I., New York, 11973.

[‡] Institute for Advanced Computation, P.O. Box 9071, Sunnyvale, California, 94086.

Research and reproduction of this report were partially supported by the Energy Research and Development Administration Contract EY-76-S-03-0326 PA #18; the National Science Foundation Grant MCS76-20019; the Office of Naval Research Contract N00014-75-C-0865; and the Army Research Office Contract DAAG-29-74-C-0034.

1. Introduction

For multi-stage linear programs with the staircase structure it has been observed that a nested decomposition (Staircase) algorithm [2],[3], can become more efficient than a direct Simplex approach. In general this tendancy increases with increasing problem size. However, even for smaller problems, the Staircase algorithm usually converges rapidly during Phase 1 and the beginning of Phase 2 before exhibiting a long "tail" towards optimality. This suggests a hybrid algorithm using Staircase initially and then switching to Simplex. Since the Staircase algorithm produces solutions which are in general nonbasic, a special interfacing procedure is required in Simplex to adopt nonbasic starting solutions.

Computational experience indicates that the hybrid algorithm can be an efficient technique for medium size problems.

2. The Staircase Algorithm

We consider the linear programming problem of minimizing

$$\sum_{t=1}^{T} c_t x_t$$

subject to

$$
\begin{aligned}
A_1 x_1 &= d_1 \\
B_{t-1} x_{t-1} + A_t x_t &= d_t, \ t = 2, \ldots, T \qquad (1) \\
x_t &\geq 0, \ t = 1, \ldots, T
\end{aligned}
$$

where c_t is $1 \times n_t$, x_t is $n_t \times 1$, A_t is $m_t \times n_t$, B_t is $m_{t+1} \times n_t$ and d_t is $m_t \times 1$ in dimensions.

In [2] and [3], the Staircase algorithm is developed from an application of nested decomposition to (1). The original problem is replaced by a sequence of smaller, independent subproblems coordinated through price and proposal communication (in the sense of Dantzig and Wolfe [1]) between adjacent subproblems. The algorithm seeks an optimal coordination (if one exists) which in turn determines an optimal solution to the original problem.

This is done in three phases. Phase 1 seeks an initial feasible solution to (1) or shows that none exists. Phase 2 seeks an optimal solution to (1) or shows that it is unbounded from below. Throughout Phase 2, primal feasibility is maintained implicitly. A Phase 3 procedure is required to reconstruct a feasible solution. This is normally done at optimality. Dual feasibility is also maintained during Phase 2 so that a lower bound on the objective is available as an optimality criterion.

It has been observed in [3] that relative to a direct simplex approach the Staircase algorithm becomes more efficient with increasing problem size. However, the threshold problem size differs considerably for different classes of Staircase problems. Moreover, even for smaller problems, the Staircase algorithm usually converges rapidly during Phase 1 and the beginning of Phase 2 before exhibiting a relatively long "tail" of convergence towards optimality.

3. A Hybrid Algorithm

The above observation suggests using Staircase to obtain a near-optimal solution and then switching over to Simplex. Hence, a Hybrid algorithm will be:

Step (i): Phase 1 of Staircase
until feasibility.

Step (ii): Phase 2 of Staircase
until objective is within p % of best lower bound where p is user supplied. ($p=\infty$ implies skipping of Step (ii)).

Step (iii): Phase 3 of Staircase
for current feasible solution

Step (iv): Staircase-Simplex Interface
from solution in Step (iii) to basic feasible solution.

Step (v): Phase 2 of Simplex.
until optimality.

Step (iv) is necessary because the solution obtained in Step (iii) is in general nonbasic. The relative amount of work in this extra step determines how well the Hybrid algorithm can combine the advantages of its components.

4. The Staircase-Simplex Interface

Assuming similar data structure for Staircase and Simplex, the actual amount of data transfer required for the switching would be negligible. It remains to start Simplex efficiently given a feasible but generally nonbasic solution.

Procedures for deriving basic feasible solutions from non-basic feasible solutions are not new. In fact such procedures,, usually called 'BASIC', are incorporated in many commercial mathematical programming systems (see e.g. [4]). Furthermore they have been used previously in ad hoc methods to partition structured problems, such as Staircase problems. The general idea is to manually or heuristically partition the right hand side (resource) vector among the submodels (time periods or divisions) which are then solved independently. Assuming a feasible partition was chosen, a non-basic feasible solution will then be available. This non-basic solution (names and values of the variables not at bound, together with bound information) is then used as a starting point for the entire undecomposed problem, and the BASIC procedure is used to derive a basic feasible solution with at least as good a value as the non-basic solution.

Although the BASIC proceedure has been implemented many times, the methodology does not seem to have been published. We therefore give an outline of the method (there are several variations).

Consider the problem of

$$\begin{array}{ll} \text{minimizing} & cx \\ \text{subject to} & Ax=b \\ & x\geq 0 \end{array}$$

with $\bar{x}$ a nonbasic feasible initial solution.

Let $J = \{j|\bar{x}_j>0\}$ and set x_j, $j\epsilon J$ nonbasic at a temporary bound (TB) of $\bar{x}_j$ with appropriate modification of the right hand side. Then proceed as follows:

Step (0) Start with all logical (or artificial) basis.

Step (1) Stop if $J=\emptyset$. Price only $x_j, j\epsilon J$ for reduced cost d_j.

Step (2) If $d_j < 0$, make x_j basic by increasing above $\bar{x}_j$, i.e. treating TB as a lower bound. If $d_j \geq 0$, make x_j basic by decreasing below $\bar{x}_j$, i.e. treating TB as an upper bound.

Step (3) Remove TB for x_j and j from J.

Return to Step (1).

Note that with this procedure the logical basis in Step (0) will be feasible at zero value since $\bar{x}$ is a feasible solution. Therefore, initially many x_j, $j \in J$ will enter the basis without changing their values from $\bar{x}_j$. Much of the computation in pricing and pivot selection can be avoided if we first scan J and pivot into the basis any x_j on any row with a nonzero entry in the updated x_j column and a zero entry in the updated right-hand-side. This "crash" procedure is equivalent to introducing the largest independent subset of x_j, $j \in J$ into the basis before setting the remainder to temporary bounds. Experiments have shown that this is essential to the efficiency of the interface.

5. Computational Experience

The experimental codes are written in FORTRAN for the CDC 7600 at Brookhaven National Laboratory. SIMPLEX is based on the revised simplex code LPM1 with product form of inverse (cf [5]). STAIRCASE is the implementation described in [3], also based on LPM1. HYBRID is based on STAIRCASE, SIMPLEX and the interface procedure in Section 4.

Computational experience with three small- to medium-size problems are reported in Tables 1 and 2 and Figure 1. It is observed that HYBRID can be used to combine the advantages of STAIRCASE and SIMPLEX.

6. Conclusion

Since the performance of special purpose algorithms is uaually highly problem dependent, the flexibility provided by a hybrid algorithm would be useful in a truly versatile Staircase decomposition system. Furthermore, basic solutions are usually preferable from a practical point of view, since fewer activities are "active". The advantage of the hybrid algorithm we propose over the ad hoc use of partitioning and the BASIC procedure, is that the Staircase algorithm appears to be successful in producing good feasible solutions relatively quickly and does not rest on unreliable user partitioning of the problem. Our preliminary computational results would indicate that the hybrid approach may be one of the more successful for the notoriously stubborn class of Staircase models.

PROBLEM / DIMENSIONS	SC205	SCFXM1	SCFXM2
PERIODS	3	4	8
ROWS	206	331	661
COLUMNS	409	788	1575
NONZEROS	758	2943	5890
% DENSITY	0.90	1.13	0.57

Table 1

Dimensions of Test Problems

CODE \ CPU TIME (SEC) \ PROBLEM		SC205	SCFXM1	SCFXM2
SIMPLEX	Phase 1	0.00	7.91	45.65
	Phase 2	3.84	3.89	29.06
	Total	3.84	11.80	74.71
	Relative	1.00	1.00	1.00
HYBRID	STAIRCASE	1.48	4.65	14.50
	INTERFACE	0.86	1.44	12.70
	SIMPLEX	0.82	3.39	41.72
$p=\infty$	Total	3.16	9.48	68.92
	Relative	0.82	0.80	0.92
HYBRID	STAIRCASE	2.34	5.32	16.38
	INTERFACE	0.92	1.06	10.27
	SIMPLEX	0.75	3.06	21.17
$p=25$	Total	4.01	9.44	47.82
	Relative	1.04	0.80	0.64
HYBRID	STAIRCASE	3.01	7.64	26.11
	INTERFACE	0.92	1.15	10.20
	SIMPLEX	0.84	3.01	13.42
$p=10$	Total	4.77	11.80	49.73
	Relative	1.24	1.0	0.67
STAIRCASE	Phase 1	1.07	4.17	11.58
	Phase 2	3.92	9.04	33.98
	Phase 3	0.48	1.18	2.95
	Total	5.47	14.39	48.51
	Relative	1.42	1.22	0.65

Table 2

Solution Times of the Test Problems

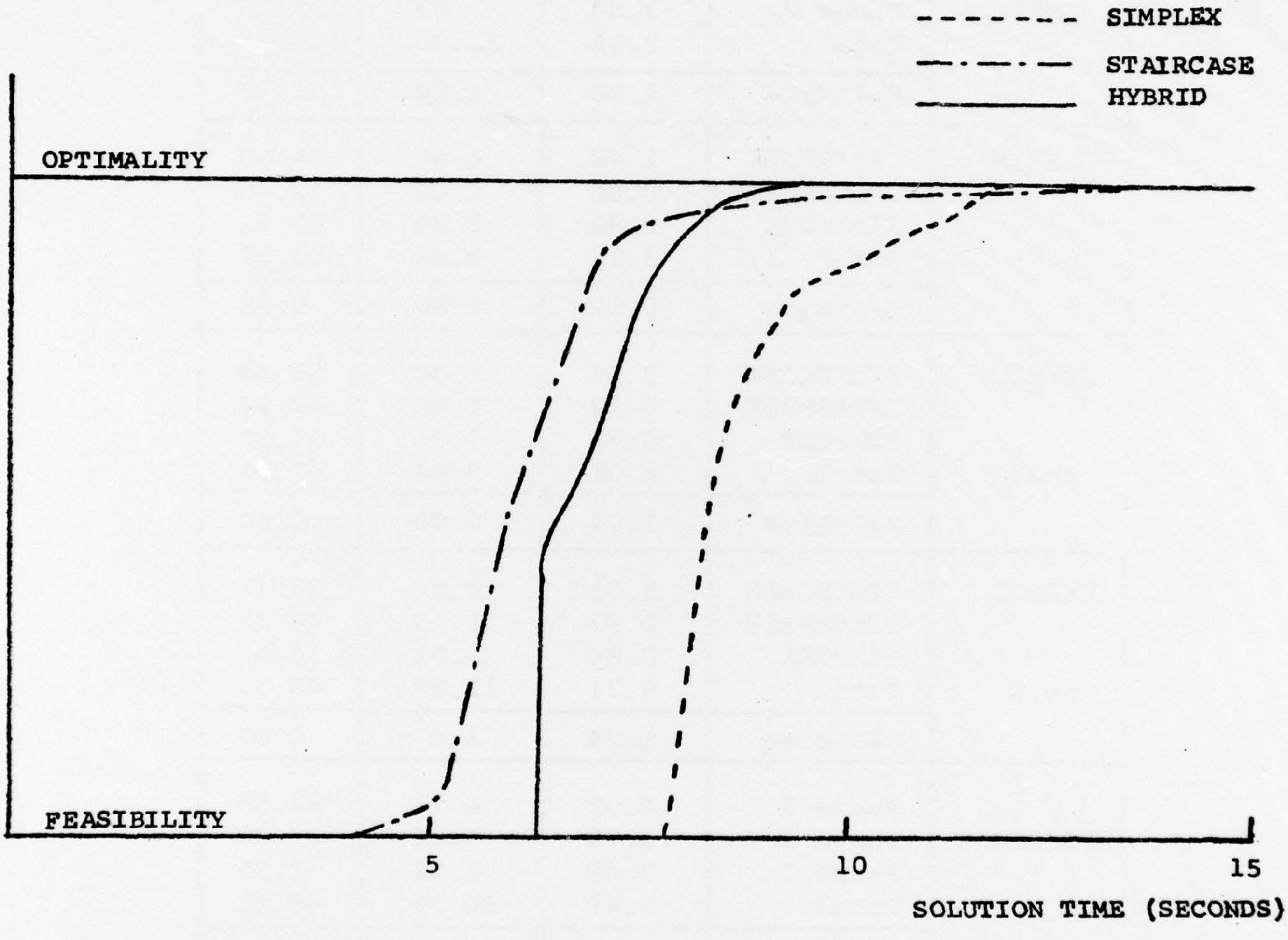


Figure 1. Solution history of Problem SCFXM1.

References

[1] Dantzig, G. B., and P. Wolfe, "Decomposition principle for linear programs", Operations Research 8 (1960), 101-111.

[2] Ho, J. K., and A. S. Manne, "Nested decomposition for dynamic models", Mathematical Programming 6 (1974), 121-140.

[3] Ho, J. K., "Implementation and application of a nested decomposition algorithm," Proceedings of the Bicentennial Conference on Mathematical Programming (in press).

[4] "Mathematical Programming System/360 Version 2, Linear and Separable Programming - User's Manual", IBM Corporation, H20-0476-2, Third Edition, October 1969.

[5] Orchard-Hays, W., Advance Linear Programming Computing Techniques, McGraw-Hill Book Company, New York, 1968.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 14 SOL-77-27

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): 6 A Hybrid Approach to Multi-Stage Linear Programs.

5. TYPE OF REPORT & PERIOD COVERED: 9 Technical Rep[redacted]t.

6. PERFORMING ORG. REPORT NUMBER: SOL 77-27

7. AUTHOR(s): 10 James K. Ho [redacted] John A. Tomlin

8. CONTRACT OR GRANT NUMBER(s): 15 N00014-75-C-0865, DAAG29-74-C-0034

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Department of Operations Research (SOL)
Stanford University
Stanford, CA 94305

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: NR-047-143

11. CONTROLLING OFFICE NAME AND ADDRESS:
Operations Research Program,
Office of Naval Research
Arlington, VA 22217

12. REPORT DATE: 11 Sep[redacted]77

13. NUMBER OF PAGES: 12 12 14P.

14. MONITORING AGENCY NAME & ADDRESS:
U.S. Army Research Office
Box CM, Duke Station
Durham, N.C. 27706

15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Linear Programming
Multi-Stage Models
Decomposition

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This paper presents a hybrid algorithm for multi-stage linear programs arising from time-phased or dynamic models. The hybrid computation is based on a nested decomposition algorithm and the revised simplex method. Initial computational experience is reported.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

408 765 JOB